Uma Abordagem para Geração de Séries Temporais de Mobilidade Urbana Baseada em Aprendizado Profundo

Iran F. Ribeiro[1], **Gabriel Simoura**[1], **Heitor S. Ramos**[2], **Giovanni Comarela**[1], **Vinícius F. S. Mota**[1] *

[1] Departamento de Informática – Universidade Federal do Espírito Santo
Vitória – Brasil
{iran.ribeiro, gabriel.simoura}@edu.ufes.br, {gc,vinicius.mota}@inf.ufes.br

[2]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais
Belo Horizonte – Brasil
ramosh@dcc.ufmg.br

Abstract. *One of the major challenges in the gathering and dissemination of urban mobility data remains on the fact that it contains information that can compromise users' privacy. An alternative to tackle this problem is the generation of synthetic datasets that may preserve the characteristics of the real data. This work evaluates such synthetic generation of time series based on urban mobility by using a classical statistical model and deep learning algorithms, such as Generative Adversarial Networks (GANs). We compare these time series against the original data by visual and quantitative analysis. Results showed that the models based on deep learning generate time series data with the same characteristics as the original dataset.*

Resumo. *Um dos grandes desafios na coleta e divulgação de dados de mobilidade urbana está no fato de que esses dados possuem informações que podem comprometer a privacidade dos usuários. Uma alternativa a esse problema é a geração de dados sintéticos que possam preservar as características dos dados reais. Este trabalho analisa a eficácia da utilização de um modelo estatístico clássico e propõe o uso de algoritmos de aprendizado profundo, como as Redes Generativas Adversarias (GANs, em inglês) para geração de séries temporais baseadas em dados de mobilidade urbana. As séries geradas foram comparadas com os dados reais por meio de uma análise visual e uma análise quantitativa. Os resultados mostraram que os modelos baseados em aprendizado profundo são capazes de gerar dados com as mesmas características dos dados reais.*

1. Introdução

O desempenho de novas tecnologias e protocolos em redes móveis, como redes móveis sem fio, redes veiculares e redes *ad hoc*, está relacionado à mobilidade dos participantes que a compõem. A mobilidade pode ser avaliada por meio de modelos de mobilidade

*O presente trabalho foi realizado com apoio financeiro da FAPES, FAPEMIG, CNPq, São Paulo Research Foundation (FAPESP) - grant #2020/05121-4 e Grant #2018/23011-1, e Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior – Brasil(CAPES) - Código 001. Ademais, este trabalho foi viabilizado pelo termo de cooperação técnica 004/2018, entre a Secretaria Municipal de Segurança Pública de Vitória-ES e a UFES.

sintéticos e/ou dados de mobilidade reais (*traces*) [Mota et al. 2014]. Modelos sintéticos permitem escalabilidade e repetibilidade mas são limitados em relação ao realismo dos dados utilizados. Por outro lado, *traces* contêm informações sobre a mobilidade de um conjunto de participantes da rede durante um período de tempo. Um *trace* pode conter a trajetória dos participantes, seja por meio de uso de posicionamento via satélite (GPS) ou transição entre estações bases de uma rede sem fio, a cada período [Zheng et al. 2009, Malandrino et al. 2018], ou pontos de partida e chegada[1]. Há também *traces* de contatos entre indivíduos, como encontros em um evento [Ribeiro et al. 2020].

Contudo, dados de mobilidade reais podem conter lacunas, não permitem variabilidade, além da dificuldade na coleta. Ademais, a divulgação desses dados geralmente é limitada por questões de privacidade. Visando sanar as deficiências supracitadas do uso de dados reais e dos modelos sintéticos, alguns trabalhos geram modelos sintéticos utilizando propriedades estatística do cenário desejado. Um exemplo é a geração de simulação de trânsito de veículos em uma cidade [Uppoor et al. 2013].

Um dos grandes desafios para a geração de dados de mobilidade urbana é que, na maioria dos casos, o comportamento desses dados possuem relações internas que variam em função do tempo. Em outras palavras, estes dados podem se tratados como uma série temporal, ou seja, um conjunto de observações que apresentam dependências entre os instantes [Brockwell et al. 2016]. Nesse sentido, os trabalhos que tratam do problema de geração de dados de mobilidade urbana, como [Gupta et al. 2018, Song et al. 2019, Jauhri et al. 2020, Zhang et al. 2020], o fazem com abordagens específicas, dificultando a generalização à outros tipos de cenários não abordados nos trabalhos.

Este trabalho apresenta um estudo de técnicas para a geração de dados sintéticos a partir de bases de dados de mobilidade reais que podem ser tratadas como séries temporais. Com isso, espera-se que que os modelos gerados a partir de dados reais possam garantir a privacidade dos usuários, adicionem variabilidade aos dados e gerem dados com as mesmas características dos reais.

Para isto, analisa-se a eficácia da utilização de modelos clássicos para predição e geração de dados como o ARIMA (*Auto-Regressive Integrated Moving Average*) e Redes Generativas Adversarias (GANs, na sigla em inglês) [Goodfellow et al. 2014] para reprodução de modelos de mobilidade urbana baseada em séries temporais. As GANs são baseadas em Redes Neurais para otimizar o treinamento de modelos generativos e tem o foco de treinar modelos que permitam gerar dados com características semelhantes aos reais. Na literatura, as GANs têm sido utilizadas majoritariamente no campo de visão computacional e para dados estático, entretanto, algumas adaptações visam a geração de dados de mobilidade urbana [Esteban et al. 2017, Yoon et al. 2019]. Nesse sentido, as GANs, enquanto conseguem reproduzir dados de mobilidade com eficiência, fornecem também bons níveis de privacidade aos dados [Qu et al. 2020].

Enquanto modelos clássicos são capazes de simular bases de dados mais simples, dados de mobilidade urbana podem necessitar de uma abordagem que não seja limitada à modelagem de relações lineares. Por este motivo, este trabalho propõe o uso de aprendizado profundo para geração de dados sintéticos a partir, e com as mesmas propriedades, das séries temporais dos dados reais. A abordagem proposta, diferente de trabalhos anteri-

[1] https://www.capitalbikeshare.com/system-data

ores, possibilita que cenários distintos de mobilidade possam ser gerados, tendo-se, assim, técnicas mais generalistas para a geração de dados sintéticos de mobilidade urbana.

Para isso, utilizou-se duas bases com características distintas: *BikeSharing*, uma base de dados aberta com informações sobre a locação de bicicletas compartilhas em cidades americanas; *Trânsito Vitória*, uma base de dados da prefeitura de Vitória com informações sobre o trânsito em tempo real. Considerando que essa última base é proprietária, os dados sintéticos representando o trânsito de Vitória é uma contribuição produto deste artigo. As demais contribuições deste artigo são sumarizadas a seguir:

- Demonstra que modelos clássicos de previsão e simulação de séries temporais, como o ARIMA, têm eficácia limitada para modelar mobilidade urbana;
- Propõe o uso de aprendizado profundo, avaliando as GANs, e demonstra que estas podem gerar dados sintéticos com propriedades dos dados reais;
- Demonstra que é possível generalizar a geração de dados de mobilidade urbana ao utilizar-se GANs que tratam de séries temporais;
- Como produto final, pode-se gerar novos dados de mobilidade (variabilidade) a partir de *traces* de mobilidade reais;

O restante do artigo está organizado como segue. A Seção 2 apresenta a fundamentação teórica, com destaque ao modelo ARIMA e as redes generativas adversárias. A Seção 3 discute os trabalhos relacionados. As bases de dados utilizadas neste trabalho são descritas na Seção 4. A metodologia para geração e validação dos modelos é apresentada na Seção 5. Por fim, a Seção 7 conclui este artigo e discute os trabalhos futuros.

2. Fundamentação Teórica

Nesta seção apresenta-se uma visão geral dos modelos de modelagem de séries temporais que foram utilizados para a geração de dados de mobilidade urbana, com suas principais características e limitações.

2.1. ARIMA

O modelo ARIMA, (Auto-Regressive Integrated Moving Average) refere-se a um modelo estatístico linear bastante conhecido na análise de séries temporais [Zhang 2003]. Ele é uma generalização de um outro modelo também bastante conhecido, o ARMA (*AutoRegressive Moving Average*), e é utilizado em séries temporais não-estacionárias, ou seja, uma série cujos dados oscilam sobre uma média que pode variar ao longo do tempo.

A definição de um modelo ARIMA usa os parâmetros p, d e q, onde p é o número de parâmetros auto-regressivos, d o número de diferenciações para tornar a série estacionária e q o número de parâmetros utilizado nas médias móveis. Esse parâmetros normalmente são definidos pela metodologia de Box & Jenkins [Box et al. 2015].

Apesar de seu bom desempenho na modelagem e previsão de séries temporais, o ARIMA possui algumas limitações como, por exemplo, ser mais eficiente em bases de dados pequenas [Montgomery and Hines 1980]. Ainda assim, é um bom modelo para se realizar as primeiras análises dos dados, indicando se um modelo não linear, por exemplo, uma Rede Neural Recorrente, precisa ser ou não utilizado.

Embora o ARIMA seja comumente utilizado para a previsão de novas observações de séries e temporais, também pode ser utilizado para a simulação de novas séries, em

que ajusta-se um modelo a uma série temporal e simula-se uma nova série com base nos parâmetros do modelo que melhor se ajustam à série. Dessa forma, é possível comparar o desempenho do ARIMA com modelos generativos, como as GANs, por exemplo.

2.2. *Generative Adversarial Networks*

Generative Adversarial Networks (GANs) é um *framework* que visa otimizar o treinamento de modelos generativos de maneira não-supervisionada por meio de um "jogo" de min-max [Goodfellow et al. 2014]. No jogo, são treinadas simultaneamente e por tempo indeterminado, duas redes neurais que competem entre si (por isso o nome *adversarial*): um Gerador ($G(z;\theta_g)$), um perceptron multicamadas que irá gerar dados falsos com base em entradas aleatórias $p_z(z)$ e um Discriminador ($D(x;\theta_d)$), outro perceptron multicamadas que classificará a qualidade dos dados gerados, considerando uma base de dados reais x. Nas redes, z, θ_g e θ_d significam, respectivamente, o espaço latente dos dados (entradas aleatórias, como distribuições normais), os parâmetros para o perceptron que define G e os parâmetros para o perceptron que define D. O objetivo de G é gerar amostras de dados cuja distribuição se aproxime tanto da distribuição real $p_{data}(x)$, que D não consiga distinguir dos reais. Assim, ao mesmo tempo que D é treinado para distinguir entre os dados oriundos de G e os reais, G terá seus pesos atualizados considerando o *feedback* fornecido por D. A Figura 1 apresenta um diagrama da estrutura básica de uma GAN.

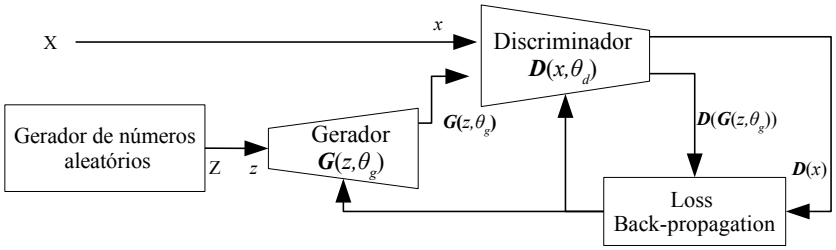


Figura 1. Arquitetura de uma GAN

Apesar dos seu bom desempenho, o *framework* proposto por [Goodfellow et al. 2014] apresenta algumas desvantagens durante o treinamento como o fato de, geralmente, não ser possível inferir a qualidade dos dados gerados sem analisá-los visualmente durante cada etapa de treinamento. O gerador pode produzir dados que são estatisticamente indistinguíveis dos reais, mas que não fazem sentido para um observador humano.

3. Trabalhos Relacionados

Desde o seu surgimento, as GANs têm sido amplamente utilizadas em estudos da área de visão computacional [Isola et al. 2017, Ledig et al. 2017, Zhu et al. 2017, Brock et al. 2018], principalmente aqueles cujas bases de dados representam imagens. Similarmente, alguns estudos que abordam a geração de dados de mobilidade urbana utilizando GANs também seguiram essa linha, como [Jauhri et al. 2020] que desenvolveram um modelo para a geração de *datasets* realísticos de pedidos de carona baseado em dados de serviços de carona de 4 cidades dos Estados Unidos. Os autores mapearam os dados contendo o ponto de partida, ponto de chegada e tempo em uma sequências de imagens, de maneira que, a cada instante de tempo, tem-se um *snapshot* da configuração dos pedidos de carona naquele instante. Em [Zhang et al. 2020], os autores modelaram as trajetórias feitas por

veículos em Beijing, China, modelando os dados como um *grid* de 32×32, em que as entradas e saídas dos carros em cada *grid* foram contabilizadas a cada 30 minutos.

De maneira similar, [Song et al. 2019] propõe a modelagem de trajetórias entre *clusters* de indivíduos baseado em dados de sistema de posicionamento global (GPS, em Inglês). Aplicando o algoritmo K-*means*, os dados são agrupados em 4 grandes grupos baseados na localização e densidade de distribuição dos indivíduos, permitindo a identificação de diferentes padrões de mobilidade entre e intra grupos. Durante o treinamento, os dados são tratados como se fossem um conjunto de imagens, permitindo que algumas técnicas já conhecidas de GANs fossem utilizadas para otimizar o treinamento.

Similarmente, [Gupta et al. 2018] descrevem um modelo de geração de trajetórias socialmente aceitáveis em ambientes com muita movimentação. No referido trabalho, a expressão "socialmente aceitável" indica uma trajetória de um indivíduo que, simultaneamente, seja fisicamente possível e respeite o espaço físico de indivíduos próximos.

Diferente dos trabalhos anteriores, [Lei et al. 2019] apresentam um modelo genérico de geração de *datasets* de mobilidade. Combinando uma GAN com uma *Graph Convolutional Network* (GCN), os autores modelam os dados de mobilidade como uma rede móvel dinâmica, em que a GCN possibilita a identificação das principais características do grafo oriundo dessa rede e, internamente, utilizam uma Rede Neural *Long Short-Term Memory* (LSTM), para capturar os padrões de mudanças da topologia do grafo ao longo do tempo. Esse modelo, por exemplo, poderia ser usado em diversos cenários em que a mobilidade pode ser modelada como grafos dinâmicos.

Embora eficientes para a geração de dados de mobilidade urbana, os modelos anteriormente citados possuem potencial limitado de serem usados para outros tipos de dados de mobilidade. Por exemplo, modelos treinados para simulação de trajetórias apenas conseguem gerar esse tipo de informação. Entretanto, os dados de mobilidade urbana, são, na maioria dos casos, altamente dependentes do tempo. É de se esperar, por exemplo, que o movimento de carros em uma determinada rua seja maior durante horários de picos e que exista uma diferença clara entre os dias úteis e finais de semana.

Assim, considerando que é factível modelar tais dados como uma série temporal, seria possível também a utilização de GANs com capacidade de geração de séries temporais para a simulação desses dados. Nesse sentido, tem-se alguns modelos de GANs conhecidos, C-RNN-GAN [Mogren 2016], RGAN [Esteban et al. 2017] e TimeGAN [Yoon et al. 2019]. A principal diferença entre os modelos citados é a arquitetura da GAN. A C-RNN-GAN é basicamente uma GAN clássica em que as duas redes neurais são uma Rede Neural Recorrente (RNN, em Inglês), mais especificamente, uma LSTM. Em seu trabalho, [Mogren 2016] utilizou a GAN proposta para gerar composições musicais com base em diversas bases de dados de diferentes compositores. A RGAN é similar à C-RNN-GAN, mas pode gerar categorias associadas à cada observação da série temporal.

Por fim, a TimeGAN é um modelo mais complexo, que além do Discriminador e Gerador possuem uma rede *Embbedding* e uma *Recovery*, que têm a função de mapear as características e espaço latente dos dados, permitindo que a rede adversária aprenda a dinâmica interna dos dados via representações de baixa dimensão [Yoon et al. 2019].

Deste modo, considerando as limitações das GANs atuais para a geração de mobilidade urbana, este trabalho analisa métodos de geração de dados sintéticos a partir de

Tabela 1. Colunas da base de dados *BikeSharing* de 2011 a 2019

Start date	Start station	End station	Bike number
2011-01-01	14th & V St NW	Calvert & Biltmore St NW	W01210

Tabela 2. Colunas da base de dados Trânsito Vitória

uuid	city	type	street	eventDate
****	Vitória	JAM	R. 2	2019-12-02 17:17:00

bases de dados reais. Considerando que dados de mobilidade podem ser modelados como uma série temporal, usar GANs que gerem séries temporais, como proposto neste artigo, ao invés de modelaram uma característica específica de mobilidade urbana, possibilita que mais tipos de dados de mobilidade possam ser modelados.

4. Bases de dados

Nesta seção descreve-se as base de dados que foram utilizadas durante os experimentos realizados neste artigo. Utilizamos uma base de dados contendo informações sobre o compartilhamento de bicicletas, em que modelou-se o número de bicicletas alugadas por hora, e uma base de dados de uma rede social de motoristas, em que modelou-se o número de carros nas ruas à cada 30 min. As bases estão melhores descritas à seguir.

4.1. BikeSharing

É uma base de dados que contém informações sobre a alocação de bicicletas em 7 cidades dos Estados Unidos (Washington, Arlington, Alexandria, Montgomery, Prince George's County, Fairfax County e City of Falls Church) entre 2011 e 2019, através do serviço de locação de bicicletas *Capital Bikeshare*. A base de dados de 01-01-2011 a 31-12-2012 foi obtida de [Fanaee-T and Gama 2013] e o número de biciletas já estava contabilizado.

Já os dados de 01-01-2013 a 31-12-2019 foram coletados do site da *Capital Bikeshare*[2]. No site, de 2012 a 2017, os dados anuais estão em um mesmo arquivo compactado, separados por quadrimestre e, entre 2018 a 2019, estão organizados, quadrimestralmente, em arquivos distintos. Algumas das colunas dessa base podem ser vistas na Tabela 1. Como nessas colunas não há informação sobre o número de bicicletas alugadas, usamos as colunas *Start date* e *Bike number* e computamos o número de bicicletas alugadas no intervalo de 1 hora.

4.2. Trânsito Vitória

Consiste em uma base de dados composta por diversas informações sobre o trânsito da região da Grande Vitória, no Espírito Santo. Essa base de dados é resultado de uma parceria entre prefeitura de Vitória com o aplicativo de trânsito para dispositivos móveis *Waze*. No *Waze*, os usuários reportam eventos atípicos nas vias, por exemplo, acidentes, alagamentos ou engarrafamentos. Além disso, o aplicativo mensura os níveis de congestionamento e contabiliza o número aproximado de carros nas vias congestionadas.

A prefeitura de Vitória recebe as informações em tempo real, em intervalos de 5 minutos. Ressalta-se que esta é uma base de dados esparsa, pois utiliza informação

[2]https://www.capitalbikeshare.com/system-data

Tabela 3. Sumarização da base de dados Trânsito Vitória

Registros	Uuids	Ruas	Cidades	Intervalo
1695712	89198	864	6	2019/12/02 - 2019/03/17

inseridas pelos usuários e contém informações de uma via apenas quando esta apresenta congestionamento. Deste modo, alguns horários não apresentam registros de eventos. A Tabela 2 apresenta as principais informações presentes na base de dados. Maiores detalhes sobre a base de dados Trânsito Vitória podem ser obtidas em [Thomé et al. 2020].

Na Tabela 2, as colunas representam: a identificação única do evento (*uuid*), a cidade (*city*) onde o evento foi reportado, o tipo de evento (*type*), o tipo da rua (*street*), nome da rua (*name*) e horário (*eventDate*) em que o evento foi reportado. Como não há informações sobre o número de carros nessa tabela, realizou-se a contabilização por meio da utilização da coluna *uuid*. Assim, assume-se que, em determinada rua, cada evento único é reportado pelo mesmo usuário. Dessa forma, em um intervalo de tempo é possível obter-se um número mínimo de carros nas ruas. Para especificar a rua, usou-se a coluna *street* e com a coluna *eventDate* seleciona-se um intervalo de tempo específico.

Além disso, na Tabela 3 é possível observar algumas informações importantes da base dados. A base de dados se inicia no dia 17 de Março de 2019 e termina em 02 de Dezembro do mesmo ano. No total, há 1.695.712 registros correspondentes à 89.198 eventos únicos, registrados em 6 cidades e 864 ruas distintas.

5. Metodologia para geração e validação dos modelos

Esta seção descreve a metodologia para geração, validação dos modelos e dos experimentos realizados, com informações sobre o ambiente de execução, os melhores parâmetros de cada modelo e as métricas utilizadas para a avaliação dos mesmos. A base de dados pública usada, as bases sintéticas geradas e os códigos para visualização dos dados estão disponíveis no *github*[3].

5.1. Ambiente de execução

Todos os experimentos foram executados na plataforma do *Google Colaboratory*[4]. Essa é uma plataforma que oferece uma máquina virtual Linux com 12 GB de memória RAM e mais de 50 GB de armazenamento. Além disso, a plataforma possibilita que os códigos sejam executados tanto em GPU (Tesla T4), por um período limitado de horas que pode variar dependendo do uso, ou CPU. Para os modelos RGAN e C-RNN-GAN é preciso fazer o *downgrade* de algumas bibliotecas, como o tensorfolow, ou executar os códigos utilizando a versão 2.7 do Python.

5.2. Pré-processamento dos dados

A maioria dos modelos de redes neurais exige algum tipo pré-processamento dos dados antes do treinamento. Além de deixar o treinamento mais rápido, algumas transformações nos dados podem garantir que os modelos de fato capturem as características dos dados. Assim, inicialmente aplicamos, a todos os dados, uma técnica bem conhecida na área de

[3]https://github.com/ifribeiro/deep_mobility

[4]https://colab.research.google.com/

Tabela 4. Parâmetros para a base de dados *BikeSharing*

	bs	**lr**	**hd**	**Épocas**
C-RNN-GAN	28	.0001	100	≈ 50
RGAN	28	.1	25	≈ 3000
TimeGAN	28	.0005	24	≈ 5000

Tabela 5. Parâmetros para as bases de dados Rua I, II e III

	bs	**lr**	**hd**	**Épocas**
C-RNN-GAN	50	.0001	100	≈ 300
RGAN	50	.1	32	≈ 3000
TimeGAN	50	.0005	72	≈ 4000

aprendizado de máquina, que consiste em transformar dados que variam de 0 a qualquer número positivo, para o intervalo [0,1].

Para o modelo RGAN, foi preciso que criássemos uma organização específica para os dados, um dicionário com a estrutura: {'**samples**':{ 'train':[],'test':[], 'vali':[]}, '**labels**':{'train':[],'test':[],'vali':[ ]}}, onde *'samples'* contém aos dados divididos em treino, teste e validação e *'labels'* contém as classes dos dados (caso possuam alguma).

Além disso, foi preciso que realizássemos a contabilização do número de carros nas ruas selecionadas da base de dados *Trânsito Vitória*. Como há um número considerável de ruas na base de dados, utilizamos apenas as 3 primeiras com os maiores registros de eventos reportados. Dessa forma, obteve-se o número de carros em cada rua em um intervalo de 30 minutos, resultando em 3 bases de dados, aqui denominadas como Rua I, Rua II e Rua III, representando cada uma das ruas selecionadas.

Para as GANs, os dados precisam estar organizados em n amostras, t intervalos de tempo e m variáveis (n, t, m). A base de dados *BikeSharing* possui 3287 amostras, 24 intervalos de tempo e 1 variável. As três bases provenientes da base *Trânsito Vitória* possui, respectivamente, 250, 230 e 257 amostras, com 48 intervalos de tempo, e 1 variável.

5.3. Hiperparâmetros

Nos experimentos feitos, buscamos identificar quais parâmetros que apresentavam os melhores resultados para os modelos avaliados. Para o ARIMA, como verificou-se que todas a bases usadas eram estacionárias, definiu-se $d = 0$ e valores diferentes para os parâmetros p e q. Por fim, utilizamos a biblioteca pmdarima[5] do Python que automaticamente identifica a combinação de parâmetros que melhor se ajustam aos dados.

Para as GANs, a análise visual dos dados, mostrou que, por exemplo, na base de dados *BikeSharing*, os dados se repetem à cada 24 horas e, assim, um valor razoável para o lote de treinamento (bs) (Tabela 4) poderia ser de 1 mês (28 dias). As bases Rua I, II e III não possuem padrões tão evidentes e não foi possível utilizar a mesma lógica. Para a escolha dos outros parâmetros, definiu-se valores distintos para cada um deles, treinou-se o modelo e avaliou-se os resultados usando-se as métricas definidas na Seção 5.4.

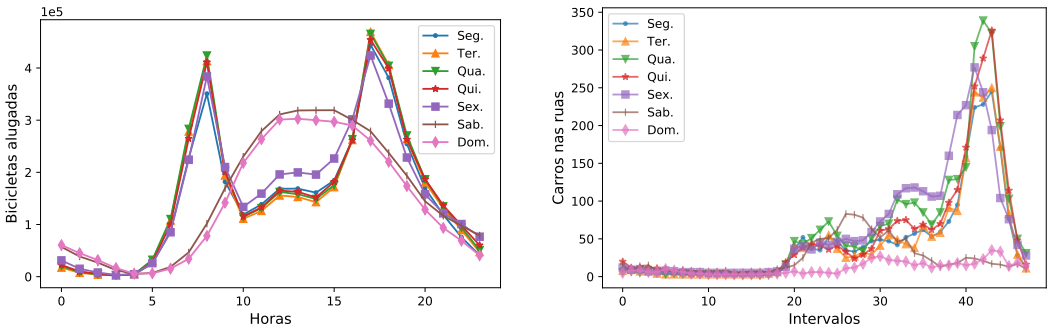
[5]https://alkaline-ml.com/pmdarima/

Dessa forma, as Tabelas 4 e 5 apresentam melhores parâmetros para as bases de dados *BikeSharing* e as bases Rua I, II e III, onde *bs*, *lr*, *hd* são, respectivamente, o tamanho do lote (*batch size*), taxa de aprendizado (*learning rate*) e dimensões escondidas (*hidden dimensions*). Como os modelos possuem dezenas de parâmetros, exibimos apenas os que mais influenciam o desempenho do treinamento dos modelos.

Ressalta-se que, como as GANs podem ter resultados levemente diferentes para os mesmos parâmetros, o valor do parâmetro Épocas representa um número aproximado de iterações em que é possível obter bons resultados para os modelos. Além disso, o comportamento dos modelos de acordo modifica-se os parâmetros é basicamente o mesmo de uma rede neural. Por exemplo, diminuir o valor de *lr* faz com que o modelo demore mais para ajustar seus pesos durante o treinamento e um *bs* muito grande reduz a capacidade de generalização dos modelos [Keskar et al. 2016]. No caso das GANs, isso pode resultar em dados sintéticos com pouca variabilidade.

5.4. Avaliação dos modelos

A avaliação de GANs, diferente de outros modelos baseados em aprendizado profundo, normalmente exige uma análise visual periódica do comportamento dos modelos. Nesse sentido, para os campos de visão computacional, como modelos para geração de imagens, a comparação visual das imagens sintéticas com a real é uma das principais formas de avaliação da eficiência dos modelos. Em séries temporais, essa forma de avaliação, embora possível, necessita de uma escolha adequada de como a comparação será feita.



(a) BikeSharing (b) Rua I (Trânsito Vitória)

Figura 2. Soma por intervalo para as bases *BikeSharing* (a) e Rua I (b)

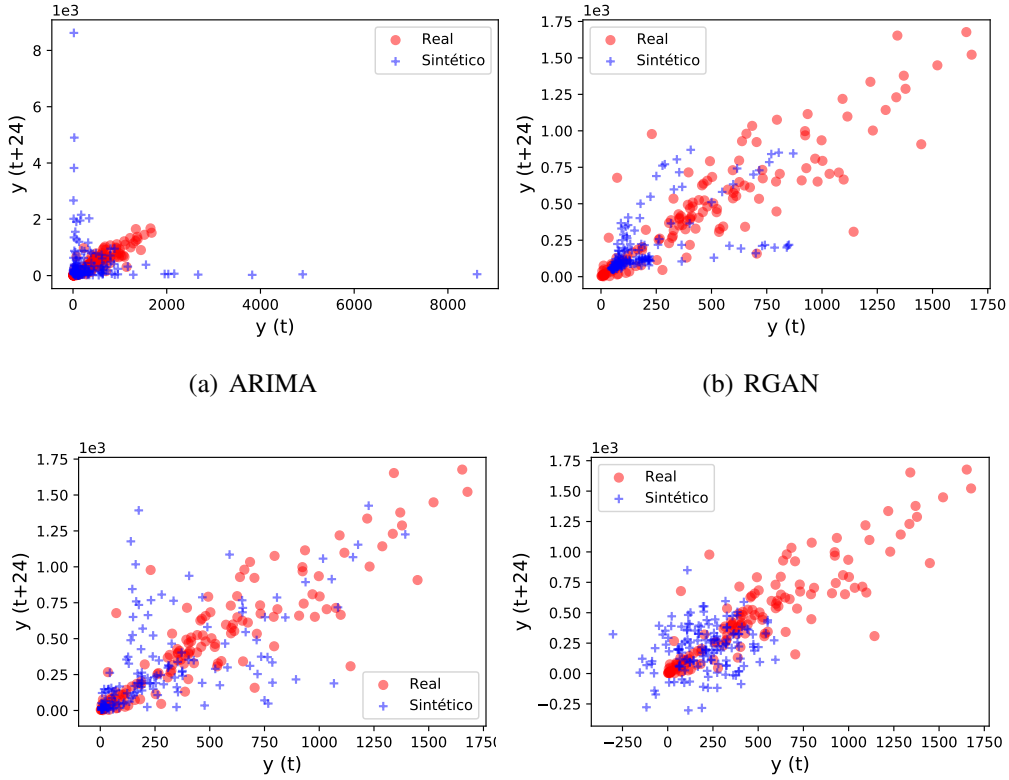
Nesse sentido, buscou-se métricas que permitissem uma avaliação qualitativa dos dados, por meio da comparação visual dos dados sintéticos gerados e uma avaliação quantitativa, que pudesse ser aplicada nos modelos treinados e possibilitassem a mensuração da eficiência dos modelos em gerar dados sintéticos que possuam as mesmas relações temporais que os reais. Assim, as métricas utilizadas são descritas à seguir:

- **Correlação Serial**: É um gráfico de dispersão em que é possível identificar as correlações internas entre um conjunto de observações de uma série temporal e suas observações futuras. Basicamente gera-se um gráfico de $S[t]$ versus $S[t+lag]$ em que *lag* representa o intervalo de observações que se deseja avaliar. Em nosso caso, esse valor é de 24 (BikeSharing) e 48 (Ruas I, II e III).

- **Soma por intervalo**: como temos informações de dia/hora sobre os dados, notamos que uma boa forma de avaliar qualitativamente os dados seria agrupá-los por cada dia da semana e somar os valores em cada horário/intervalo. Por exemplo, dados gerados através da base de dados *BikeSharing* devem apresentar padrões semelhantes aos observados na Figura 2(a). Com picos aproximadamente às 8 e 17hs nos dias da semana e com picos entre as 12 e 15hs nos finais de semana.
- **Análise dos resíduos**: Para avaliar quantitativamente eficiência dos modelos utilizados, analisamos os resíduos de 1000 amostras dos dados sintéticos de cada modelo, para cada base utilizada. Para isso, ajustamos um modelo ARIMA para obter os resíduos dos dados gerados pelo ARIMA, e utilizamos um modelo com duas camadas de LSTM para obter os resíduos dos dados sintéticos das GANs.

6. Avaliação dos modelos sintéticos gerados

Os resultados dos experimentos podem ser vistos a seguir. Na Figura 3 analisamos a correlação serial dos dados sintéticos em relação aos reais para a base de dados *BikeSharing*, em um intervalo de 24 horas. Utilizamos essa base pois ela possui maior variabilidade entre os dados e selecionamos, aleatoriamente, uma parte dos dados (reais e sintéticos) para a análise. Nas figuras, observa-se que os dados reais apresentam uma tendência linear de crescimento, indicando que há uma correlação positiva entre observações presentes e futuras. Os modelos utilizados deveriam capturar essas correlações.



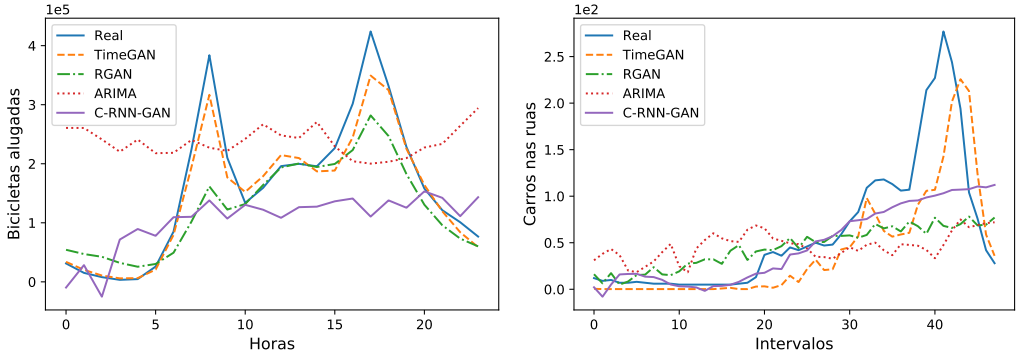
(a) ARIMA (b) RGAN

(c) TimeGAN (d) C-RNN-GAN

Figura 3. Correlação serial para a base de dados *BikeSharing*

Esperava-se que os dados sintéticos apresentem-se de maneira semelhante aos reais, caso tenham sido produzidos por um bom modelo gerador. Entretanto, pela Figura

3(a) infere-se que o ARIMA não conseguiu captar as características dos dados corretamente, já que podemos notar pontos (em azul) muito distantes dos dados reais (em vermelho). Nas Figuras 3(b) e 3(c), contudo, notamos uma aglomeração dos pontos sintéticos aproximadamente onde concentram-se os pontos reais, mostrando que as GANs, no geral, captaram melhor as relações temporais entres os dados. Entretanto, como pode ser visto na Figura 3(d), a C-RNN-GAN gerou dados negativos, o que não deveria ocorrer.



(a) BikeSharing (b) Rua I (Trânsito Vitória)

Figura 4. Número de bicicletas alugadas em 24hs (a) e o número de carros na rua à cada intervalo de 30 min (b), calculadas a partir das sexta-feiras

Para avaliar se os modelos capturaram os padrões de comportamento dos dados, agrupamos os dados gerados pelos modelos em cada um dos dias da semana, e somamos os valores em cada hora do dia. Para isso utilizamos as base de dados *BikeSharing* e a base Rua I, que apresentam padrões mais evidentes do que as outras bases analisadas.

Assim, na Figura 4 tem-se a comparação da soma dos dados da base *BikeSharing* com os dados sintéticos dos diferentes modelos analisados, durante as sextas-feiras. Nota-se que os dados gerados pela TimeGAN apresentaram padrões muito parecidos com os reais, com picos por volta das 8 e 17hs horas (Figura 4(a)) e, na Figura 4(b), por volta do intervalo 40. A RGAN capturou os padrões da base *BikeSharing*, porém não tão bem quanto a TimeGAN. Por outro lado, não conseguiu modelar bem os dados da Rua I.

A C-RNN-GAN, apesar de não conseguir modelar bem os dados da base *BikeSharing* (inclusive apresentando valores negativos), conseguiu capturar a tendência de crescimento da base Rua I, com menos carros nas ruas entre os intervalos 0 e 20 e um crescimento entre os intervalos 20 e 40. Contudo, nota-se que não foi possível capturar a decrescimento exibido pelos dados reais após o intervalo 40.

Tabela 6. Análise dos resíduos dos modelos. Melhores resultados em negrito

	BikeSharing	**Rua I**	**Rua II**	**Rua III**
ARIMA	.000 (.530)	.000 (1.534)	.000 (.100)	.000 (.748)
TimeGAN	**.007 (.158)**	.314 (.131)	**.002 (.072)**	.014 (.081)
RGAN	.025 (.154)	**.005 (.032)**	.316 (.094)	.056 (.050)
C-RNN-GAN	.078 (.185)	.005 (.037)	.300 (.0187)	**.001 (.019)**

Por fim, o ARIMA não conseguiu modelar bem os dados tanto na base *BikeSharing* quanto na base Rua I. De fato, pelas Figuras 4(a) e 4(b), nota-se que os dados gerados

pelo ARIMA apresentam um padrão linear, com oscilações no número de carros alugados e de carros nas ruas, mas não sendo possível identificar a similaridade com os dados reais.

Os resíduos dos ajustes dos modelos podem ser vistos na Tabela 6 (médias e desvios padrão, entre parênteses). Os melhores resultados são mostrados em negrito. Nota-se que as GANs apresentaram os melhores resultados, mesmo nas bases de dados menores. A TimeGAN mostrou-se melhor que os outros modelos em 2 das 4 bases e, apesar do ARIMA apresentar média 0 para distribuição dos resíduos para a base de dados *BikeSharing*, o alto desvio padrão indica que o modelo não consegue representar bem os dados.

6.1. Discussão

Como visto na análise dos resíduos da Tabela 6, ainda que o ARIMA apresente média 0 para os resíduos, a comparação visual dos dados reais e sintéticos da Figura 4 mostra que os padrões internos nos dados não foram corretamente aprendidos. Isso ocorre porque o ARIMA não consegue aprender dependências temporais muito distantes entre si e é um indicador de que os modelos precisam ser avaliados de diferentes formas para garantir a sua eficiência. Em contra partida, os bons resultados da TimeGAN são consequência das redes adicionais (*Embedding* e *Recovery*) presentes em sua arquitetura, que possibilitam um treinamento em representações de baixas dimensões dos dados.

Embora as bases de dados usadas neste artigo tenham características similares (são contagem de carros/bicicletas alugadas), as GANs para séries temporais poderiam ser treinadas com outros tipos de dados. Poderia-se gerar, com uma base de dados com localização GPS, os pontos mais visitados por pessoas à cada intervalo de tempo. Consequentemente, é possível identificar as trajetórias feitas pelas pessoas e também pontos de interesse. Em relação à área de redes móveis, seria possível modelar contatos feitos pelas pessoas, considerando suas localizações.

Assim, a metodologia proposta pode ser usada não apenas para dados citados neste artigo, mas também para outros tipos de cenários, inclusive para aqueles discutidos nos trabalhos que introduziram os modelos de GANs para séries temporais. A limitação, nesse caso, é que os dados precisam pertencer ao conjunto dos números reais.

Finalmente, é importante salientar que o objetivo das GANs não é gerar dados exatamente iguais aos reais. Se isso ocorresse, os dados gerados não iriam possuir variabilidade e nem privacidade. Por isso, já era esperado que os modelos não apresentassem exatamente as mesmas propriedades vistas nas análises qualitativas.

7. Considerações Finais

Neste trabalho, analisamos o desempenho de 3 modelos de GANs na geração de séries temporais de mobilidade urbana e comparamos com o ARIMA, um modelo clássico de previsão de séries temporais. Para avaliação dos modelos, utilizamos uma base de dados com número de bicicletas alugadas em 7 cidades dos Estados Unidos, e três bases com o número de carros em 3 ruas da cidade de Vitória, no Espírito Santo, Brasil.

Avaliou-se visualmente a qualidade dos dados sintéticos, comparando-os com os reais e quantitativamente, analisando-se os resíduos obtidos pelos modelos ajustados aos dados. Os experimentos mostram que o ARIMA é ineficiente na modelagem e simulação dos dados, mesmo nas bases de dados menores, não conseguindo capturar as características presentes nos dados, como por exemplo, padrões em horários de pico.

Mostramos que as GANs, principalmente a TimeGAN, conseguiram modelar bem as séries temporais, identificando os padrões e as relações internas entre os dados. Assim, com a abordagem proposta é possível gerar-se diferentes tipos de dados de mobilidade urbana, considerando que tais dados possam ser tratados como séries temporais.

Por fim, em trabalhos futuros pretende-se avaliar a metodologia proposta em outros dados de mobilidade e desenvolver um *framework* para a geração de séries temporais de mobilidade urbana que utilize uma das três GANs apresentadas neste trabalho, forneça métodos de avaliação dos dados gerados e possa ser utilizado em dados multivariados.

Referências

Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

Brock, A., Donahue, J., and Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

Brockwell, P. J., Brockwell, P. J., Davis, R. A., and Davis, R. A. (2016). *Introduction to time series and forecasting*. Springer.

Esteban, C., Hyland, S. L., and Rätsch, G. (2017). Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*.

Fanaee-T, H. and Gama, J. (2013). Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, pages 1–15.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., and Alahi, A. (2018). Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2255–2264.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134.

Jauhri, A., Stocks, B., Li, J. H., Yamada, K., and Shen, J. P. (2020). Generating realistic ride-hailing datasets using gans. *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, 6(3):1–14.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. (2016). On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690.

Lei, K., Qin, M., Bai, B., Zhang, G., and Yang, M. (2019). Gcn-gan: A non-linear temporal link prediction model for weighted dynamic networks. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 388–396. IEEE.

Malandrino, F., Chiasserini, C., and Kirkpatrick, S. (2018). Cellular network traces towards 5g: Usage, analysis and generation. *IEEE Transactions on Mobile Computing*, 17(3):529–542.

Mogren, O. (2016). C-rnn-gan: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv:1611.09904*.

Montgomery, D. C. and Hines, W. W. (1980). *Probability and statistics in engineering and management science*. John Wiley & Sons.

Mota, V. F., Cunha, F. D., Macedo, D. F., Nogueira, J. M., and Loureiro, A. A. (2014). Protocols, mobility models and tools in opportunistic networks: A survey. *Computer Communications*, 48:5 – 19. Opportunistic networks.

Qu, Y., Yu, S., Zhou, W., and Tian, Y. (2020). Gan-driven personalized spatial-temporal private data sharing in cyber-physical social systems. *IEEE Transactions on Network Science and Engineering*.

Ribeiro, I., Castanheira, L., Schaeffer-Filho, A., Cordeiro, W., and Mota, V. (2020). Caracterização de mobilidade e detecção de comunidades baseadas em tópicos de interesse. In *Anais do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 603–616, Porto Alegre, RS, Brasil. SBC.

Song, H. Y., Baek, M. S., and Sung, M. (2019). Generating human mobility route based on generative adversarial network. In *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 91–99. IEEE.

Thomé, M., Prestes, A., Gomes, R., and Mota, V. (2020). Um arcabouço para detecção e alerta de anomalias de mobilidade urbana em tempo real. In *Anais do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 784–797. SBC.

Uppoor, S., Trullols-Cruces, O., Fiore, M., and Barcelo-Ordinas, J. M. (2013). Generation and analysis of a large-scale urban vehicular mobility dataset. *IEEE Transactions on Mobile Computing*, 13(5):1061–1075.

Yoon, J., Jarrett, D., and van der Schaar, M. (2019). Time-series generative adversarial networks.

Zhang, G. P. (2003). Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175.

Zhang, H., Wu, Y., Tan, H., Dong, H., Ding, F., and Ran, B. (2020). Understanding and modeling urban mobility dynamics via disentangled representation learning. *IEEE Transactions on Intelligent Transportation Systems*.

Zheng, Y., Zhang, L., Xie, X., and Ma, W. (2009). Mining interesting locations and travel sequences from gps trajectories. In *World wide web*, pages 791–800. ACM.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.